

## 8PRO128: Programmation Orientée Objet

### TP1 : une partie du Miniprojet Travail peut se faire en binôme

À remettre le 12 Février 2020<sup>1</sup>

---

#### 1. But

Permettre aux étudiants de se familiariser avec:

- La création d'applications contenant plusieurs classes;
- L'abstraction;
- L'encapsulation;
- Les constructeurs et les destructeurs.

#### 2. Travail à effectuer

Vous êtes engagé par la compagnie «Soft.inc» pour mettre en place un logiciel de gestion d'une ligue de soccer (ça peut être aussi le Hockey, je vous laisse le choix), et de simulation d'une partie de jeu (un match) entre deux équipes lors d'une rencontre (prochain TP2). Dans le cadre de ce premier travail pratique, on vous demande de créer la première partie métier (noyau) du logiciel pour assurer la gestion des clubs. Il s'agit d'offrir un certain nombre de services permettant l'ajout d'un nouveau club à une ligue, de créer, de modifier les caractéristiques d'un sportif (un joueur, un entraîneur,...), de le supprimer s'il y a lieu en affichant leurs informations, ou bien répondre à l'interrogation concernant le club ou l'entraîneur le plus titré (ayant plus de titres).

Pour cela, il faut préparer la structure (modèle) du noyau de cette partie en tenant compte des spécifications suivantes :

- a) Un **club** de Soccer (Hockey) est caractérisé par :
  - *Histoire du Club* est une description textuelle de type string;

---

<sup>1</sup> Le travail doit être remis dans le dossier de remise sur moodle et aucun travail n'est accepté par courriel.

- **Couleur du club** de type string (ou bien une image);
- **Année de création** de type date ou autre;
- **Effectif** : une liste de type Joueur;
- **un Palmarès** de type **Palmares**. Un Palmarès est un titre (une coupe, ou un championnat) obtenu par un club à une certaine date donnée;
- **Stade du Club** de type **Stade**. Un stade est caractérisé par une capacité (nombre maximal de personnes qu'il peut accueillir en fonction des places disponibles), la qualité du terrain de foot (gazon ou tartan), le nom du stade de type string, et son adresse ou lieu de type String ;
- **Ville du Club** de type string et l'adresse du Club de type string;
- **Staff Technique** est une liste de **Personne**. Chaque personne du staff est caractérisée par son nom de type string, Age de type int, et son rôle qui peut être : président (directeur ou manager), secrétaire, recruteur, infirmier-spécialisé. Dans le cadre de ce type, l'héritage n'est pas obligatoire. Vous pouvez réduire le rôle en une simple chaine de caractères ou un numéro faisant référence aux entités concernées en question

b) Chaque **joueur** est caractérisé par les informations suivantes :

- **Nom et prénom** de type string ;
- **Taille** du joueur de type float;
- **Poids** de type float ;
- **Ville de Naissance** de type string;
- **Parcours du Joueur** est une liste de type « **Parcours** ». Un parcours est caractérisé par le nom du Club où il avait joué ainsi que la date de son passage au sein de ce club. Le club actuel du joueur peut faire partie du parcours.

Pour localiser un joueur (une recherche dans le système), on peut utiliser la concaténation du club actuel, suivi de son nom et de son prénom. On suppose que dans une même équipe, il n'y a pas de joueurs qui ont les mêmes noms et prénoms.

c) Un **entraîneur** est chargé de préparer physiquement, tactiquement, etc, son équipe pour les différentes rencontres. C'est un sportif ayant :

- un nom et prénom de type string
- **lieu d'obtention du Grade** de type string
- **liste de titres gagnés** de type TitreGagné. Cette liste est un ensemble de Palmarès, auquel on ajoute une propriété de type Club, précisant le club avec lequel il a gagné le titre en question.

d) vous devez créer une classe «**LigueSoccer**» ou «**LigueHokey**» qui matérialisera la

ligue en question selon votre choix d'application. Pour la structure, je vous laisse le choix de me proposer une structure de classe cohérente. Pour garder une trace de la liste des instances de classes, vous utiliserez des vecteurs. La classe «vector» est acceptable ou un autre type de conteneur.

- e) Vous devez créer une ou plusieurs classe(s) Ecran(s) qui devraient contenir en principe des méthodes « statiques » d'affichage, de saisie et de menu principal. Dans le but de ne pas mélanger d'une part, le traitement applicatif avec la présentation et d'autre part, pour faciliter le passage à l'environnement graphique lors du prochain travail pratique. À la limite vous pouvez créer des « vraies » interfaces usagers à la Windows avec un des générateurs d'interfaces.
- f) L'application doit respecter les principes d'abstraction et d'encapsulation. Le nom de la classe d'application doit être: LigueSoccerApp.

### 3. L'application doit permettre :

- 3.1 l'ajout d'un nouveau club ainsi que sa suppression;
- 3.2 l'affichage de la liste des joueurs d'un certain club;
- 3.3 La localisation de l'entraîneur le plus titré de la ligue et le club le plus titré de la ligue.

### 4. Livrables

Les livrables pour ce premier travail sont:

- 4.1 Une page de garde contenant votre nom, prénom et le code permanent;
- 4.2 Des captures d'écran d'exécution pour chacune des fonctionnalités demandées en 3.1 jusqu'à 3.3
- 4.3 L'exécutable de l'application, le code source documenté (lisible), validé de l'application (.cpp) et les fichiers (.h) dans votre compte.

NB :

- Dans le cas d'un travail en binôme, la remise d'un seul exemplaire dans un des comptes des membres du groupe est suffisant. Il faut ajouter un fichier readme.txt pour indiquer les membres du groupe et s'il y a lieu toute autre hypothèse prise en considération ou fonctionnalité supplémentaire.
- **Aucun livrable ne sera accepté par un envoi de courriel.**

***Bon succès.***